CRAYONIC PAPER

Enabling decentralized, trusted, secure & compliant digital signatures in electronic documents in enterprise & public sectors per eIDAS & GDPR regulations

Signing per eIDAS.

Crayonic Paper is free and open source mobile application. It is optimized for reading and signing of documents in PDF format. Ideally the document should conform to ISO32000 standard or even to PDF/A format to be more future proof for archiving reasons. The software can create signature per PAdES LTV standard which makes it compliant with regulations across countries (especially in EU).

Out-of-the-box the Crayonic Paper app will try to connect to a public timestamp authority (TSA) during the signing process to obtain a verifiable timestamp as part of the PAdES signature (the free use of the public TSA is limited to 5 timestamps per day), if signing offline the timestamp of the local device will be used as an untrusted source. Besides the ability to sign PDF document with the PAdES signature, the app will insert visual handwritten mark of the signatory at signature location within PDF (what-you-see-is-what-you-sign) and metadata that are related to the given signature - this metadata may contain other attributes besides those above such as the name of the signatory, email, phone number but also asymmetrically encrypted data of the handwritten signature dynamics such as pressure, speed, linear acceleration or angular velocity etc. Depending on the setup of the application and the source of certificate, the PAdES signature may reach the level of Qualified Electronic Signature.

Crayonic B.V. High Tech Campus 1E 5656 AE Eindhoven The Netherlands
KvK 67932762 VAT ID: NL857231820B01
sales@crayonic.com www.crayonic.com +31 402 310 107 +421 908 925 135

Verifying Signatures

Additionally, Crayonic Paper app also allows some of the most important signature verifications of existing digital signatures e.g. when existing signature is tapped it will notify the user if the signature is still valid (document has not been modified after the signature) and if the identity of the signatory can be linked to the trusted root certificate of certificate authority (TSP CA) or even if the timestamp of the signature can be trusted (provided from TSA or not). OCSP and CRL verifications are also available and may be executed in the background.

Secure Document Exchange

Even though reading, signing and verifying signatures are the main use cases for the application, we have decided to add few additional features that our users have found very useful during user testing. The most important of these features is the ability to securely exchange a document between devices e.g. PC-to-mobile, mobile-to-mobile, mobile-to-PC. This is achieved by first generating true random number as an AES-256 key which is then exchanged between devices out-of-band via QR codes, SMS or push messaging depending on the location of signing parties that need to exchange the document. The document is always encrypted on the device that is sending the document and only after encryption it is send via internet in P2P or assisted P2P modes.

Scanning

Scanning is the next important feature that the app can provide to an end user, allowing creation of the PDF document with the mobile device and starting its digital workflow whether it's just signing the document or sending it to another user for signature. Scanning currently supports some basic but important post processing, such as auto cropping, perspective correction, color and contrast corrections.

QR Code Reading and Setup

Finally, the app has an integrated QR code reader which serves for multiple purposes.
When QR code contains a URL to PDF document it will immediately download it and open it for reading/signing. If the QR code contains a URL pointing to valid HTML page, then the page will be displayed - this can be useful when the HTML page contains multiple links for PDFs or some type of workflow UI. The encrypted PDF can be passed via QR code using URL and AES-256 (CTR) key separated by '0x10h' character.

QR code may also contain a link to URL of an encrypted (ideally signed) ZIP package (details in Appendix). The ZIP package allows complete customization of the application itself. The ZIP will need to contain HTML5/CSS/Javascript folder for UI customization, branding and some optional business logic. Furthermore, it may contain set of trusted root certificates allowing enterprises to set custom trust anchors for the app instead of relying on default device root certificates. Besides UI, and certificates, the *Application.properties* file should be included which can define many additional security and customization options available (details in the Appendix).

Certificate Creation and Storage

The Crayonic Paper app supports creation of an end-user identity via x509 certificate (self-signed or signed by a remote trusted CA) backed by the private key generated on the mobile device in the HW key store which most modern devices now allow. The access to the private key signing operations is protected by the device authentication mechanism (password, PIN, fingerprint etc.) When the user needs to connect to a remote enterprise service, this certificate can be used for mutual SSL authentication and/or for the PAdES digital signature created by the end-user.

Because of this, Crayonic Paper supports signing of PDF documents using handwritten signature backed by three types of digital signatures:

a) **Digital Signature created by the certificate stored on the device and created for the device owner prior to signing any document. Identity is created when application start and does not find any usable x509 certificates in the app key store, this is the most common use case.**

b) **Digital Signature created with an ad-hoc identity per the metadata in the digital signature PDF form field e.g. signature of a third party other than the owner of the device. This digital signature can be created online remotely on a HSM device or locally by a certificate that is self-signed or signed by a remote CA after identity verification.**

c) **Digital Signature created by a cryptoki HW token, however since mobile devices do not support PKCS11 or similar standard to access any cryptoki HW, Crayonic Paper supports only devices that have been implemented within the app itself e.g. Crayonic Pen.**

Simple UI and Workflows

The ability to customize the main UI of the app is possible due to embedded web browser based on Chromium OSS project. This feature guarantees consistent behavior of the application UI across devices and allows implementation of custom signing/scanning workflow wizards using simple HTML and javascript. Developers familiar with the concept of hybrid mobile application development such as PhoneGap/Cordova should be able to customize Crayonic Paper to any special needs of their customers.

Use case for Crayonic Paper used with Crayonic Pen (earlier prototype) demo:
https://youtu.be/3OrOnfEpA9A